

Large Language Models for Deobfuscation: the Case of Mixed Boolean Arithmetic Obfuscation

Keywords: Deobfuscation, artificial intelligence, program analysis

Main Supervisor

Grégoire MENGUY

CEA LIST

gregoire.menguy@cea.fr

Senior Supervisor

Sébastien BARDIN

CEA LIST

sebastien.bardin@cea.fr

Obfuscation [3, 4] aims to protect software from reverse engineering. It translates a program P into a functionally equivalent program P_o , harder to analyze. While obfuscation is used to protect *Intellectual Property* and other valuable software assets, it is also used to protect malware. Thus, automated *deobfuscation* methods [11, 2, 5, 9, 10] have been proposed to cope with the quick advances in obfuscation. Given an obfuscated program P_o , the goal is to simplify it into a simpler yet functionally equivalent program P^* – ideally, P^* should be as simple as the original unprotected code P . Interestingly, one obfuscation received considerable interest from both the protection and attack communities: Mixed-Boolean Arithmetic (MBA) protection [12]. MBA translates bitvector expressions into new, more complex expressions that combine both bitwise and arithmetic operators. A wide variety of deobfuscation methods dedicated to MBA deobfuscation have been proposed, from white-box [7, 8] and black-box [9, 2], to grey-box deobfuscation [5]. White-box methods use the syntax of the code to simplify it through deduction and simplification rules. The black-box relies on program synthesis [6] to simplify highly obfuscated code snippets. Finally, grey-box approaches combine white- and black-box analyses to get the best of both worlds.

Recently, Large language models (LLMs) have shown strong capabilities in program generation and analysis. Still, only a few works have studied LLMs for deobfuscation [1]. This project aims to fill this gap, be it for white-, black- and grey-box deobfuscation. It will proceed as follows:

- Get familiar with the state-of-the-art of white-, black- and grey-box deobfuscation;
- Evaluate LLMs on MBA deobfuscation tasks. The student will focus on LLMs whose API are accessible in free access (e.g., Gemini);
- Compare LLMs against the state-of-the-art of MBA deobfuscation.

1 Expected deliverable

- A summary of the bibliography made by the student;
- A documented evaluation of the tested LLMs;
- A final report (slides) with the first two deliverables and a summary of the research and results.

2 Organization

Regular meetings will be organized with the supervisor (online or in person at CEA Nano-Innov, Saclay).

3 How to apply

To apply, students must send an email to the main supervisor (Grégoire Menguy) with the senior supervisor (Sébastien Bardin) in copy.

References

- [1] David Beste et al. “Exploring the potential of llms for code deobfuscation”. In: *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer. 2025, pp. 267–286.
- [2] Tim Blazytko et al. “Syntia: Synthesizing the Semantics of Obfuscated Code”. In: *USENIX Security*. 2017.
- [3] Christian Collberg and Jasvir Nagra. *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*. Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection, 2009.
- [4] Christian Collberg, Clark Thomborson, and Douglas Low. *A taxonomy of obfuscating transformations*. 1997.
- [5] Robin David, Luigi Coniglio, and Mariano Ceccato. “QSynth-A Program Synthesis based Approach for Binary Code Deobfuscation”. In: *BAR 2020 Workshop*. Internet Society, 2020.
- [6] Sumit Gulwani, Oleksandr Polozov, Rishabh Singh, et al. “Program synthesis”. In: *Foundations and Trends® in Programming Languages* (2017).
- [7] Jaehyung Lee and Woosuk Lee. “Simplifying Mixed Boolean-Arithmetic Obfuscation by Program Synthesis and Term Rewriting”. In: *Conference on Computer and Communications Security*. 2023.
- [8] Binbin Liu et al. “{MBA-Blast}: Unveiling and simplifying mixed {Boolean-Arithmetic} obfuscation”. In: *30th USENIX Security Symposium (USENIX Security 21)*. 2021, pp. 1701–1718.
- [9] Grégoire Menguy, Sébastien Bardin, and Bonichon et al. “Search-Based Local Black-Box Deobfuscation: Understand, Improve and Mitigate”. In: *Conference on Computer and Communications Security*. 2021.
- [10] Rolf Rolles. “Unpacking Virtualization Obfuscators”. In: *USENIX Conference on Offensive Technologies*. WOOT’09. 2009.
- [11] Jonathan Salwan, Sébastien Bardin, and Marie-Laure Potet. “Symbolic deobfuscation: from virtualized code back to the original”. In: *DIMVA*. 2018.
- [12] Yongxin Zhou, Alec Main, and Gu et al. “Information Hiding in Software with Mixed Boolean-arithmetic Transforms”. In: *Conference on Information Security Applications*. 2007.